Timer Services

Thor Design Panel 3

84K00510-110

Version 1.1
11/24/97

Software Requirements and Design Specification

1. Timer Services

1.1 Timer Services Introduction

1.1.1 Timer Services Overview

The Timer Services CSC will provide support for application access to the Coordinated Universal Time (UTC) value. Timer Services will also process commands to set the value of the Countdown Time (CDT) or Mission Elapsed Time (MET) and to place a hold on the CDT/MET. Timer Services will publish via Data Distribution the CDT/MET value, the CDT/MET status, the time of the next hold, and the time of the next resume.

Timer Services will also allow applications to request interrupts after a delay of a specified time, or at a specified UTC value, CDT value, or MET value. Timer Services will provide a stopwatch function.

1.1.2 Timer Services Operational Description

The Thor delivery will provide an initial release of Timer Services. Timer Services will be provided through a set of APIs that will allow an application to get the current UTC, ~~Countdown Time (CDT)/Mission Elapsed Time (MET),~~ request an interrupt after a specified elapsed time and request an interrupt when UTC, CDT or MET equals a specified value. ~~Timer Services will also provide CDT/MET capabilities to start, hold (immediately or at a specified time) and set CDT/MET.~~ Timer Services will also process CDT/MET commands received via Command Management. The commands supported are to start, hold (immediately or at a specified time) and set CDT/MET. CDT/MET services will be provided by Application Services APIs and by a Command Support display. Another Command Support display will support the stopwatch function.

Refer to Timer Services Data Flow Diagram, Section 1.2.4. This diagram shows a local timer server function on each C&C WS as well as a central timer server function on a CCP. The local and central timer functions are accessed through the Timer Services APIs and the Timer GUI. Application Services will interface with Command Management to route CDT/MET commands to the central timer server. This use of Command Management will provide authentication of CDT/MET commands. The central timer server maintains the CDT/MET and publishes it as a system Function Designator (FD) to data distribution at a cyclic rate. Data Distribution then places this FD into the RTCN and DCN change data streams. CDT/MET are available to applications as an FD via Application Services.

~~For Thor, T~~thh~~e~~ The Timer Services ~~CDT/MET~~ GUI's will be ~~independent of the~~ initialized from the Command Navigation Service, or from Display Services. The Stopwatch GUI will be initialized from Display Services o~~o~~nly.

1.2 Timer Services Specifications

1.2.1 Timer Services Ground Rules

- Each C&C WS, the CCP and the DDP processor clocks will be set to UTC via the Network Timing Protocol (NTP) from the IRIG-B signal it receives.
- ~~Application Services will interface directly with the local timer server for local requests.~~
- Application Services will interface with Command Management to route Countdown Time (CDT) or Mission Elapsed Time (MET) commands to the central timer server.
- Command Management will provide authentication of timer commands (i.e. only authorized users will be allowed to manipulate CDT/MET).
- The Timer Server processes will run as root, with real-time priority.
- A Backup Timing Server will not be part of the Thor delivery.

- The Timer Services GUIs will not be supported in the BASIS environment.
- Timer Services will support a maximum of fifty application interrupt timers per process.

1.2.2 Timer Services Functional Requirements

Basic

1. Timer Services will publish the CDT/MET value as a system FD.
2. Timer Services will publish the CDT/MET status (held or counting) as a system FD.
3. Timer Services will publish the CDT at which the next CDT hold is scheduled as a system FD.
4. Timer Services will publish the UTC at which the CDT will resume counting as a system FD.

CDT/MET User Interface

5. Timer Services will display the current Coordinated Universal Time (UTC).
6. Timer Services will provide a visual indication of the status of the UTC (synchronized via NTP or not synchronized).
7. Timer Services will display the current Countdown Time (CDT) or Mission Elapsed Time (MET).
8. Timer Services will provide a visual indication of the status of the Countdown time (held or counting).
9. Timer Services will allow a user to set the Countdown Time (CDT) or Mission Elapsed Time (MET) to a value.
10. Timer Services will allow a user to immediately start the Countdown Time (CDT) or Mission Elapsed Time (MET).
11. Timer Services will allow a user to start the Countdown Time (CDT) or Mission Elapsed Time (MET) at a specified Coordinated Universal Time (UTC).
12. Timer Services will allow a user to immediately place a hold on the Countdown Time (CDT) or Mission Elapsed Time (MET).
13. Timer Services will allow a user to place a hold on the Countdown Time (CDT) or Mission Elapsed Time (MET) at a specified CDT or MET .
14. Timer Services will allow a user to cancel pending Countdown Time (CDT) or Mission Elapsed Time (MET) commands.

CDT/MET and Timer Interrupt APIs

15. Timer Services will provide an API to get the current Coordinated Universal Time (UTC).
16. ~~Timer Services will provide an API to get the current Countdown Time (CDT) or Mission Elapsed Time (MET).~~
17. Timer Services will provide an API to generate an application interrupt after a delay of specified time with a granularity of one millisecond.
18. Timer Services will provide an API to generate an application interrupt when the Coordinated Universal Time (UTC) equals a specified time with a granularity of one millisecond.
19. Timer Services will provide an API to generate an application interrupt when the Countdown Time (CDT) or Mission Elapsed Time (MET) equals a specified time with a granularity of one second.
20. Timer Services will support multiple application interrupt timers per process.
21. Timer Services will provide an API to cancel an application timer interrupt request.
22. *Timer Services will provide an API to cancel all of an application's timer interrupt requests* .
23. Timer Services will process a command to set the Countdown Time (CDT) or Mission Elapsed Time (MET) to a value.
24. Timer Services will ~~provide an API~~ process a command to immediately start the Countdown Time (CDT) or Mission Elapsed Time (MET).
25. Timer Services will ~~provide an API~~ process a command to start the Countdown Time (CDT) or Mission Elapsed Time (MET) at a specified Coordinated Universal Time (UTC).
26. Timer Services will ~~provide an API~~ process a command to immediately place a hold on the Countdown Time (CDT) or Mission Elapsed Time (MET).
27. Timer Services will ~~provide an API~~ process a command to place a hold on the Countdown Time (CDT) or Mission Elapsed Time (MET) at a specified CDT or MET.

28. Timer Services ~~provide an API~~ will process a command to cancel pending Countdown Time (CDT) or Mission Elapsed Time (MET) commands.

Stopwatch User Interface

29. Timer Services will allow a user to start a Stopwatch Timer.
30. Timer Services will allow a user to stop a Stopwatch Timer.
31. Timer Services will allow a user to clear (reset to zero) a Stopwatch Timer.
32. Timer Services will allow more than one Stopwatch Timer per C&C WS.

Stopwatch API's

33. Timer Services will provide an API to start a Stopwatch Timer.
34. Timer Services will provide an API to stop a Stopwatch Timer.
35. Timer Services will provide an API to clear (reset to zero) a Stopwatch Timer.
36. Timer Services will provide an API that returns the elapsed time since the Stopwatch Timer was started.

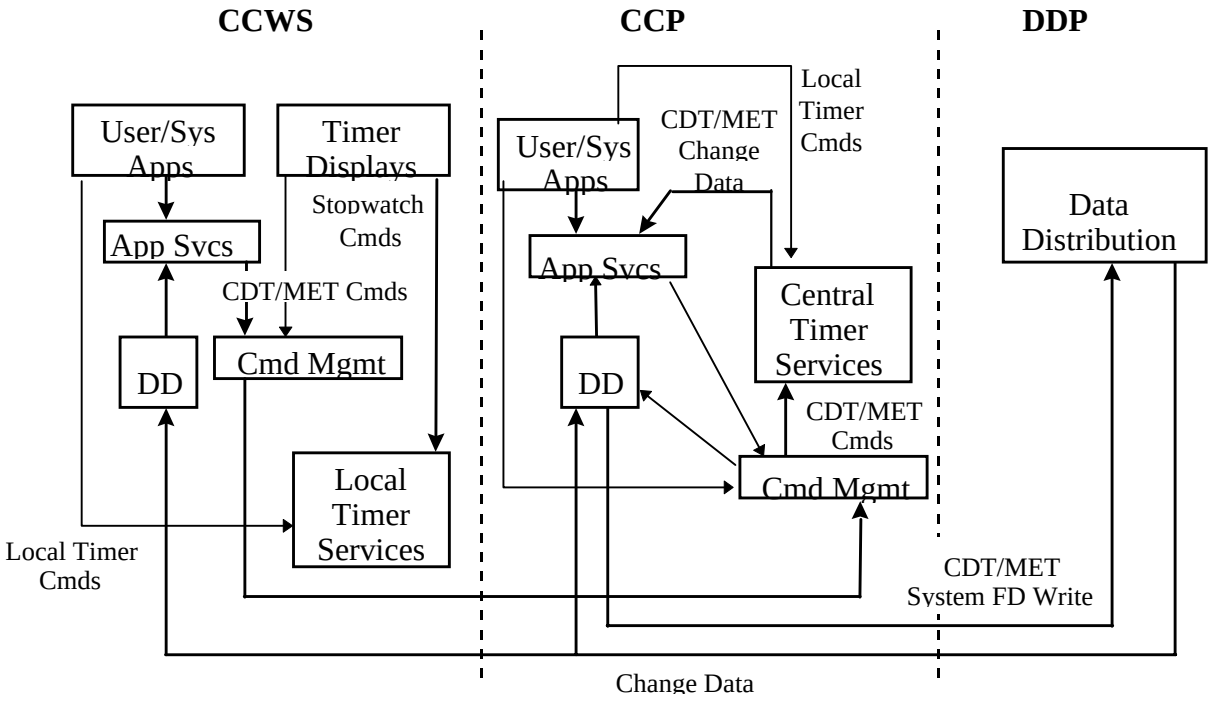
System Integrity

37. Timer Services will detect the loss of workstation synchronization to UTC.
38. Timer Services will report the loss of workstation synchronization to Subsystem Integrity.

1.2.3 Timer Services Performance Requirements

1. Timer Services must publish the CDT/MET value once per second, with a ten millisecond accuracy within ten milliseconds of the start of a second, to Application Services on the CCP.
2. Timer Services must provide notification of delay expiration or achievement of a UTC specified value with a ten millisecond precision.

1.2.4 Timer Services Interfaces Data Flow Diagrams



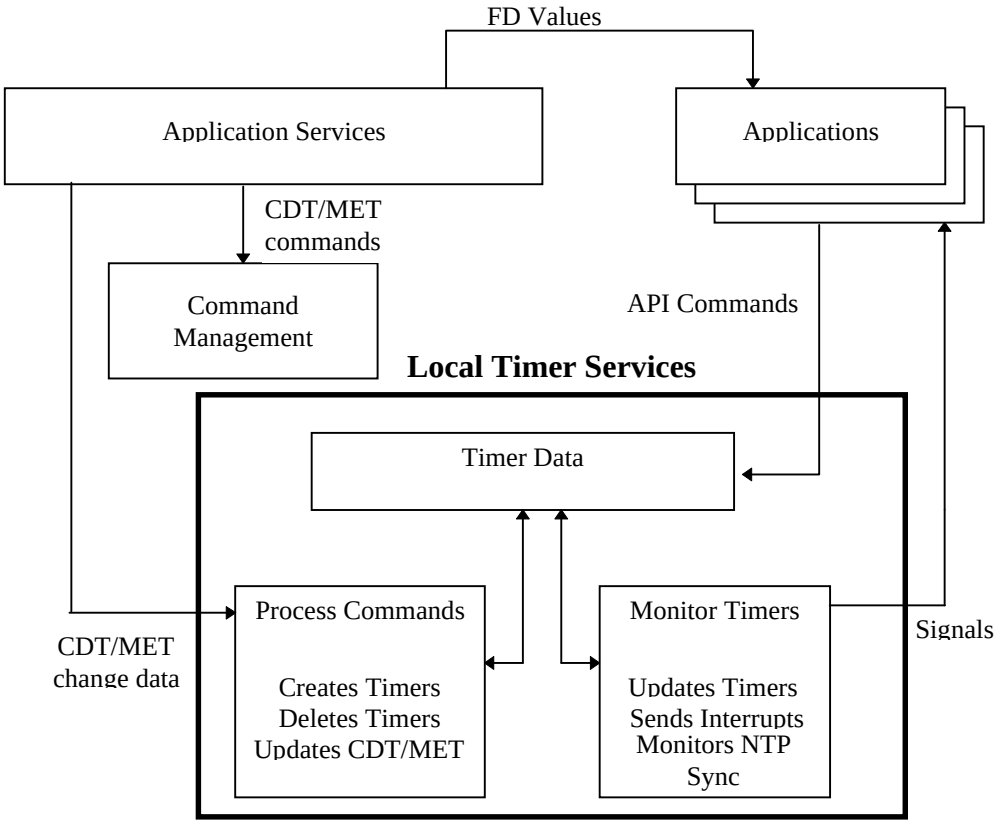
Timer Services Data Flow

1.2.5 Definitions

1. **Granularity** - smallest increment which can be specified.
2. **Precision** - degree of refinement with which a measurement is stated.
3. **Accuracy** - degree of conformity of a measure to a standard value.

1.3 Timer Services Design Specification

1.3.1 Timer Services Detailed Data Flow



Local Timer Server Architecture

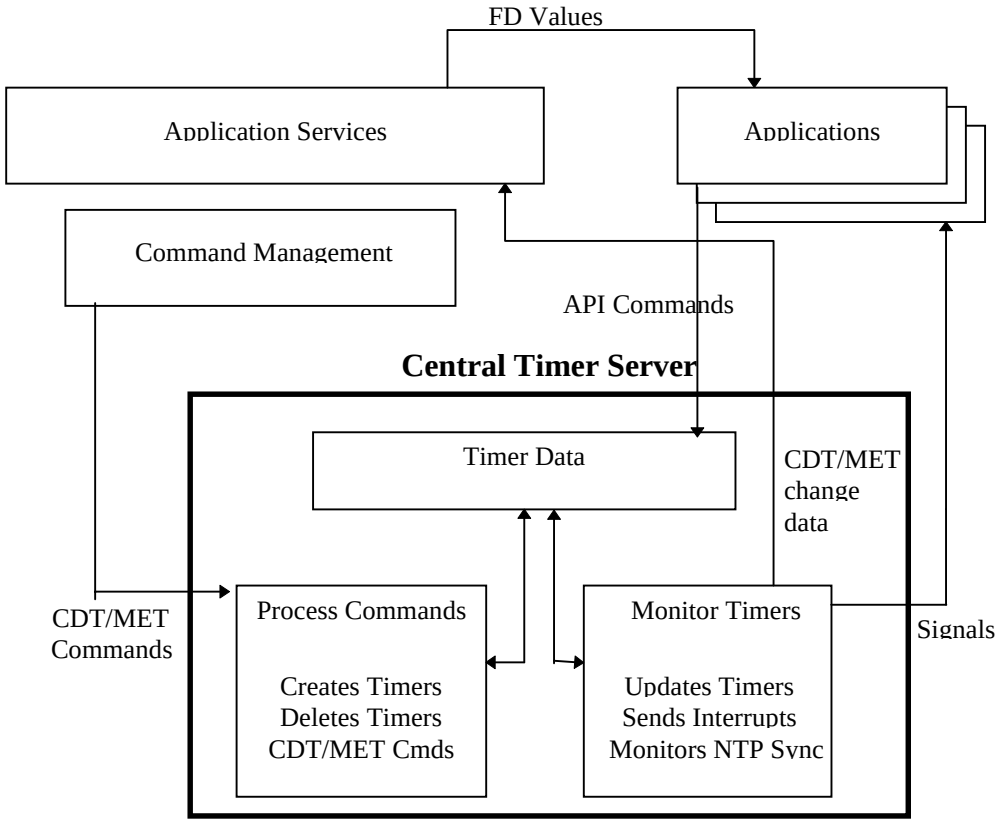
The Timer Server will consist of two main functional pieces, the Commands Process, and the Monitor Timers Process. Timer Services must process commands and update CDT/MET data asynchronously as they are received. Timer Services must also monitor timers continuously to detect timer expiration, since timers may be based on CDT. The Commands Process will listen for timer commands and update the Timer Data. The Monitor Timers Process will detect timer expiration, deliver application interrupts, and monitor NTP synchronization.

API Flow

1. Applications call the Timer Services API to create a timer.
2. The Timer Services API stores the timer information in the Timer Data area.
3. Timer Services monitors the Timer Data area until the timer expiration is detected.
4. Timer Services sends a signal to the requesting application.

CDT Flow

1. The Local Timer Server will monitor the system FD for CDT that is published by the Central Timer Server.
2. The CDT value will be stored in the Timer Data area for use in monitoring timers that expire at a specific CDT.



Central Timer Server Architecture

API Flow

The API flow will be the same as for the Local Timer Server.

CDT Flow

1. The Central Timer Server will store the CDT/MET data in the Timer Data area. This includes all values that will be published as system FDs.
2. The Central Timer Server will receive CDT/MET commands from Command Management.
3. The Timer Data area will be updated based on the command input.
4. Any changed FD values will be published via Application Services at the start of each second. The CDT/MET value will be updated and published at the start of each second.

1.3.2 Timer Services External Interfaces

1.3.2.1 Timer Services Message Formats

Message Number = TBD
Message Group = TBD

CDT set.

No inserts.

Message Number = TBD
Message Group = TBD

CDT held.

No inserts.

Message Number = TBD
Message Group = TBD

CDT counting.

No inserts.

Message Number = TBD
Message Group = TBD

CDT count will begin at I#1.

Insert #1 = string with GMT time formatted as HHMM/SS:mmm

Message Number = TBD
Message Group = TBD

CDT count will be held at I#1.

Insert #1 = string with CDT time formatted as +/- HHMM/SS

Message Number = TBD
Message Group = TBD

CDT pending count at I#1 has been canceled.

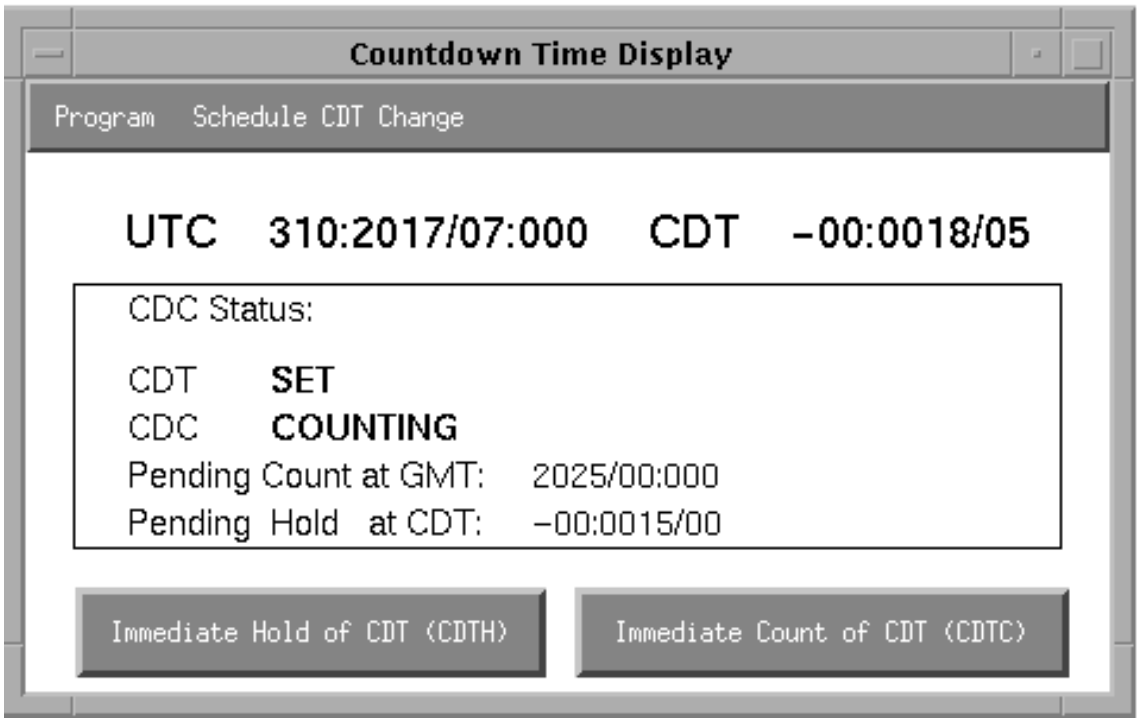
Insert #1 = string with GMT time formatted as HHMM/SS:mmm

Message Number = TBD
Message Group = TBD

CDT pending hold at I#1 has been canceled.

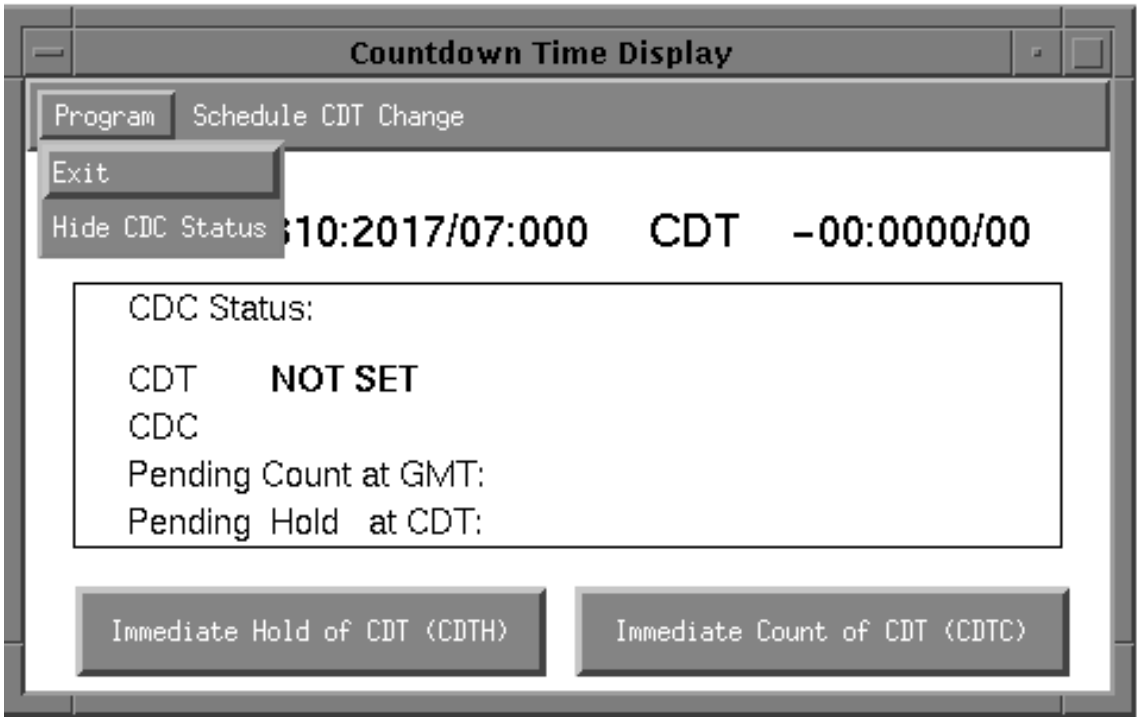
Insert #1 = string with CDT time formatted as +/- HHMM/SS

1.3.2.2 Timer Services Display Formats



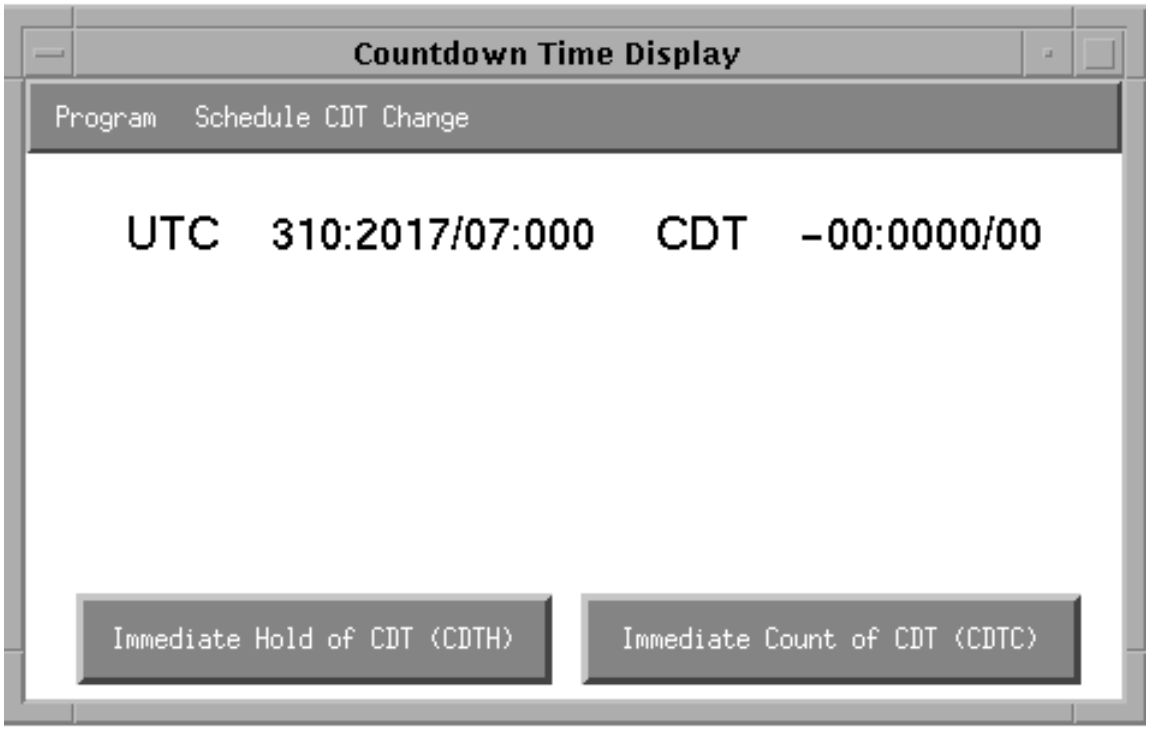
Countdown Time Display

Shown on the top line is Universal Coordinated Time and Countdown Time. In the outlined box below these two times is the Countdown Clock (CDC) Status. The first line shows whether the Countdown Time is set or not set. The second line shows whether the Countdown Clock is counting or holding. The third line displays a GMT time value when the Countdown Clock will begin counting if a pending count is set. The fourth line displays a CDT time value when the Countdown Clock will go into a hold, if a pending hold is set. The bottom two buttons are for issuing immediate commands. The left button places an immediate hold on the Countdown clock, and the right button immediately starts the Countdown Clock counting. The CCMS system command that corresponds to each action is printed on the button. For example, a Countdown Clock hold is “CDTH”, and this is printed on the “Immediate Hold of CDT” button. This connection with CCMS commands has been maintained throughout the display.



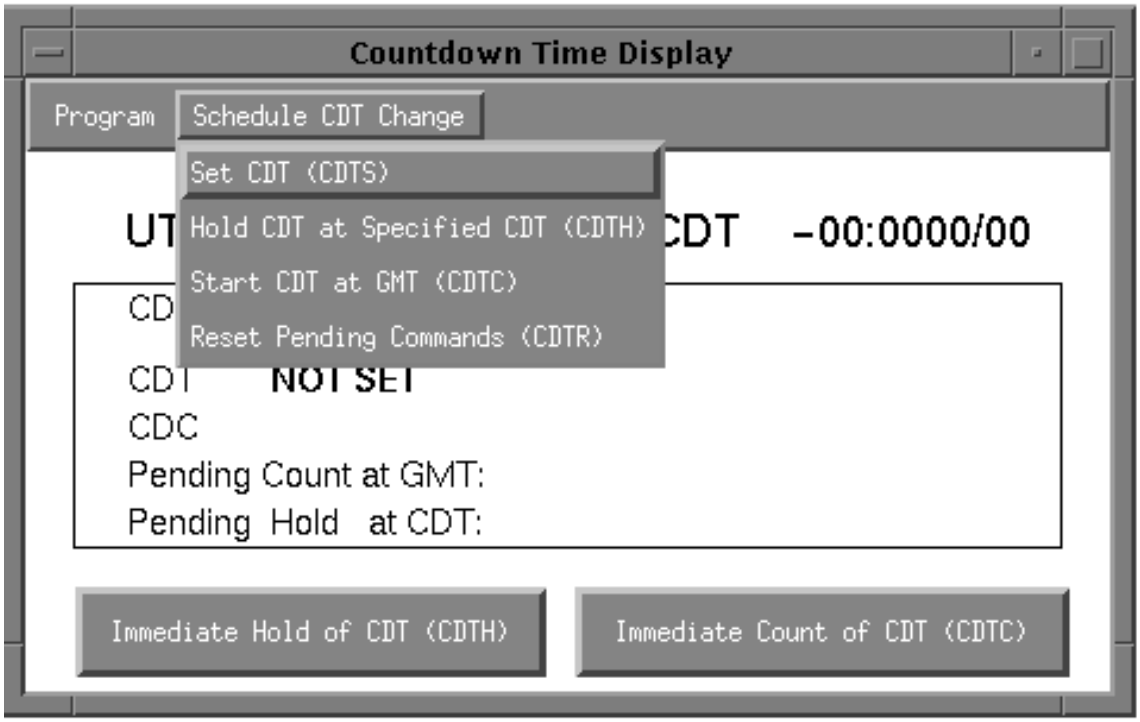
Countdown Time Display - Program Menu

The “Program” menu will have “Exit” and “Hide CDC Status” options. The “Program->Exit” option will pop up a confirmation dialog and exit the program.



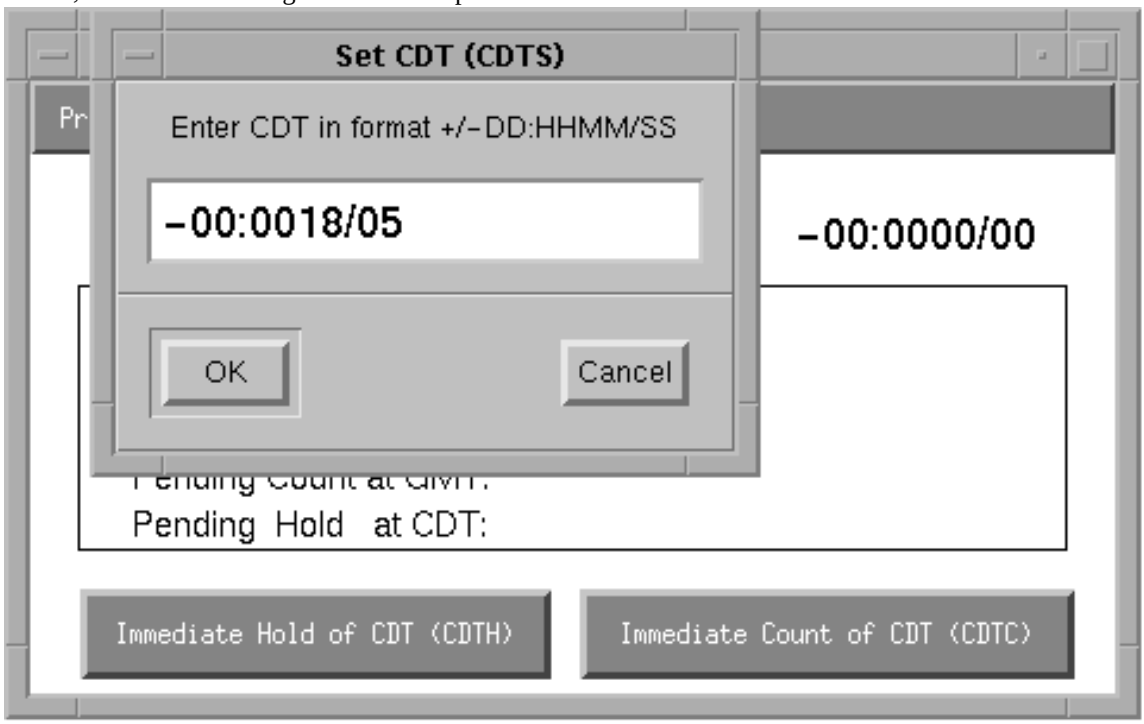
Countdown Time Display - Hiding CDC Status

The “Program->Hide CDC Status” option hides the CDC Status Box and changes the menu option to “Show CDC Status”. Selecting “Program->Show CDC Status” will make the CDC Status Box reappear.



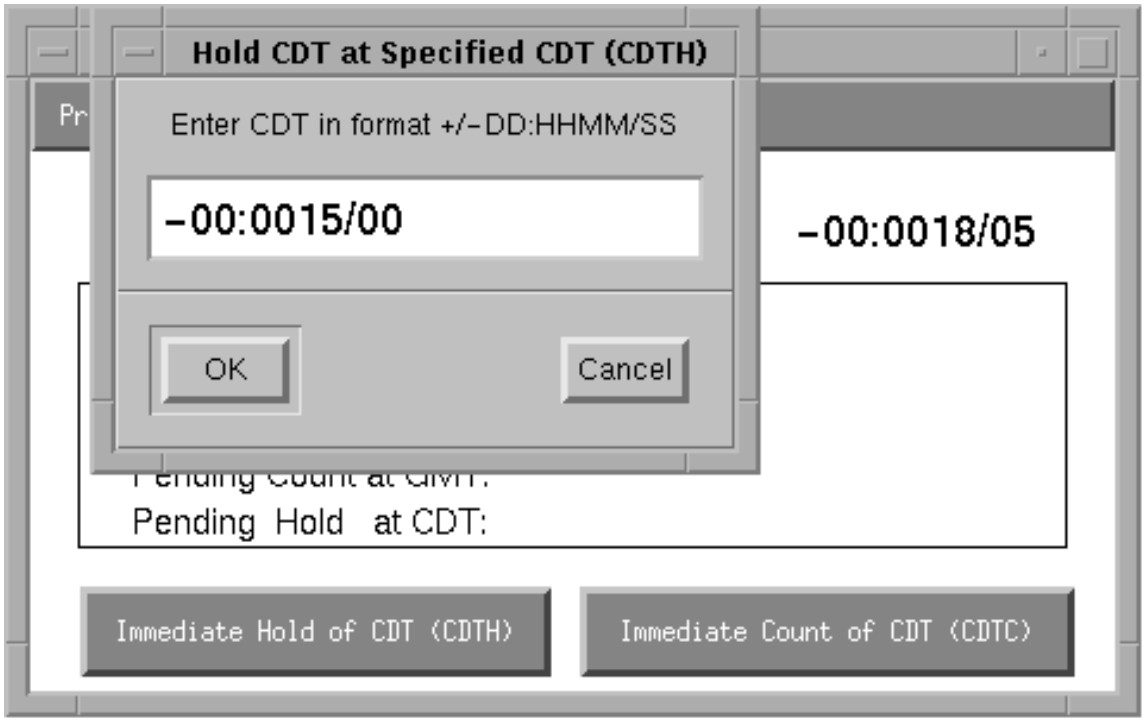
Countdown Time Display - Schedule CDT Change Menu

The “Schedule CDT Change” menu will have “Set CDT”, “Hold CDT at Specified CDT”, “Start CDT at GMT”, and “Reset Pending Commands” options.



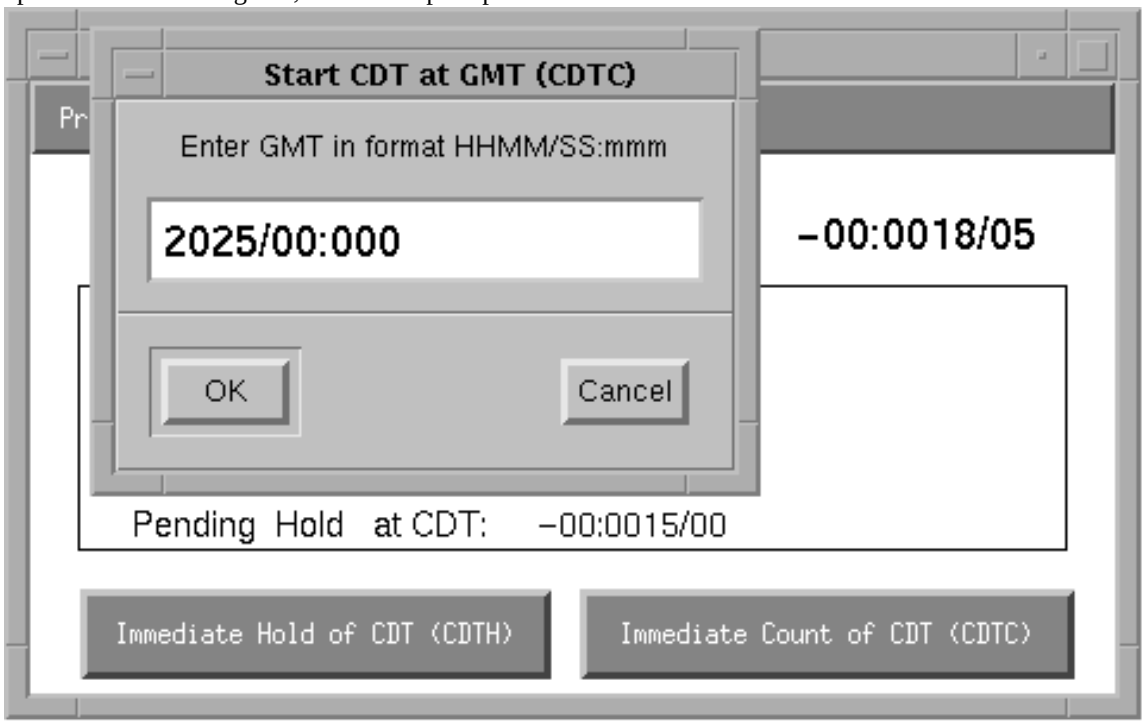
Countdown Time Display - Set CDT Dialog

The “Schedule CDT Change->Set CDT” menu option will bring up a “Set CDT” dialog box, which will prompt the user for a CDT value.



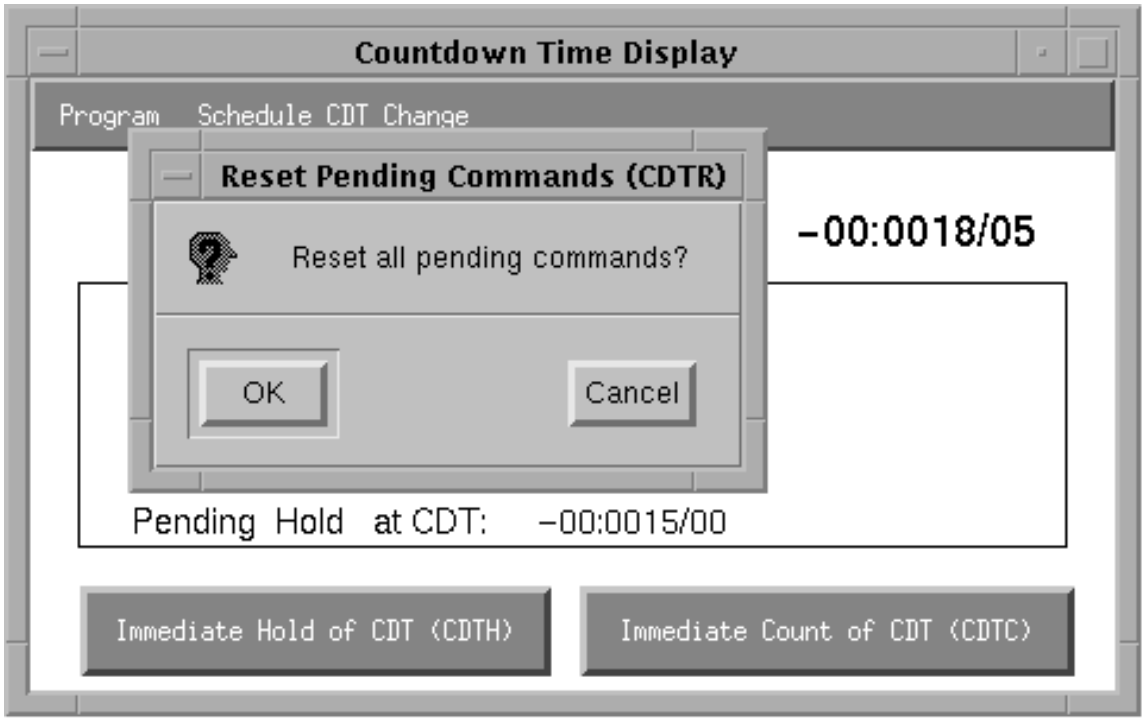
Countdown Time Display - Hold CDT at Specified CDT Dialog

The “Schedule CDT Change->Hold CDT at Specified CDT” menu option will bring up a “Hold CDT at Specified CDT” dialog box, which will prompt the user for a CDT value.



Countdown Time Display - Start CDT at GMT Dialog

The “Schedule CDT Change-> Start CDT at GMT” menu option will bring up a “Start CDT at GMT” dialog box, which will prompt the user for a GMT value.



Countdown Time Display - Reset Pending Commands Dialog

The “Schedule CDT Change->Reset Pending Commands” menu option will bring up a “Reset Pending Commands” confirmation dialog box. This command will cancel all pending hold and pending count commands, and they will be removed from the display.

Countdown Time Display - Countdown Clock Held

When the Countdown Clock is held, the Countdown Time Display will reverse the Countdown Time background and foreground colors, and will show the Countdown Clock "HOLDING" in the Countdown Clock status box. Although it is not shown here, the Countdown Time Display will change the Universal Coordinated Time foreground color to yellow when Timer Services detects the loss of workstation synchronization to UTC.

Stopwatch Timer Display

The top line of the Stopwatch Timer display shows the stopwatch time minutes, seconds, & milliseconds. The second line shows the GMT time when the stopwatch timer is started. The third line shows the GMT time when the stopwatch timer is stopped. The "Start" button will cause a "Start GMT" time to be displayed, and the stopwatch time will begin counting up. The label on the "Start" button will be changed to " Stop".

Stopwatch Timer Display - Stopwatch Started

The “Stop” button will cause a “Stop GMT” time to be displayed, and the stopwatch time will stop counting and display the final result. The label on the “Stop” button will be changed back to “Start”. The “Clear” button will set the stopwatch time, Start GMT time, and Stop GMT time to zero.

1.3.2.3 Timer Services Input Formats

This section is not applicable for Timer Services.

1.3.2.4 Timer Services Printer Formats

This section is not applicable for Timer Services.

1.3.2.5 Interprocess Communications

Timer Services will receive CDT/MET commands from Command Management via IPC messages in Payload Packet format. Timer Services will deliver response messages via the same mechanism. The packets containing CDT/MET commands must contain time data for the commands which schedule a CDT/MET hold or resume to occur at a particular time. For a UTC time, the time data will be provided in the standard Unix format of seconds since 1/1/70, and microseconds since the start of the second. For a CDT/MET time, a seconds value is all that is needed. For convenience, the payload packet for all commands will be identical.

Payload packet expected by timer services is as follows:

CC Packet header	40 bytes
seconds	4 bytes
microseconds	4 bytes

In the CC packet header the route code used will be 98. The following request IDs will be used:

set CDT/MET to a value	1
start CDT/MET	2

hold CDT/MET	3
cancel pending CDT/MET commands	4

1.3.2.6 Timer Services External Interface Calls

Timer Services will provide the following class and methods:

class **TS_localTimer**

+**set** ( ) - sets a timer with default values

+**set**(label, milliseconds, timer type, time string format, notify type) - sets the timer with the provided parameters

+**set**(label, struct timeval, timer type, time string format, notify type) - sets the timer with the provided parameters

+**destroy**() - removes the timer from the timer table

+**destroyAll**() - removes all timers created by the process from the timer table

These methods may throw the following exceptions:

sharedMemoryAttach - generated by a failure to attach to the timer table in shared memory

sharedMemoryDetach - generated by a failure detaching from the timer table in shared memory

semaphoreFailure - generated by a failure getting or releasing a semaphore

timerTableFull - generated when a timer attempts to find a slot in the timer table but none are available

unsetTimer - generated when an operation is attempted on an unset timer

class TS_stopWatch

+clear() - clears the stopwatch

+start() - starts the stop watch

+stop() - stops the stop watch

+getElapsedTime() - gets the current elapsed time

int TS_GetCurrentUTC(struct timeval *T, int *S) - retrieves the current UTC and returns it in a timeval structure. The user must have allocated the memory for T and S. TS_GetCurrentUTC returns a zero upon success and T will contain the current UTC and S will contain the status of synchronization of UTC . Upon failure T will be NULL and S is set to -1.

1.3.2.7 Timer Services Table Formats

This section is not applicable for Timer Services.

1.3.3 Timer Services Test Plan

1. The Timer Services CDT/MET Graphical User Interface will read CDT/MET, CDT/MET Status, pending hold time, and pending count time FD's. These values can be set through the same Graphical User Interface, and when the FD values change, it can be observed on the display.
2. The correct UTC, CDT/MET, and CDT status will be visually verified on the Timer Services CDT/MET Graphical User Interface. The buttons to set CDT/MET, start CDT/MET, hold CDT/MET, and cancel pending CDT/MET commands will be tested and the results will be visually verified.
3. The Timer Services CDT/MET Graphical User Interface will call the Timer Services API to get the current UTC. This can be visually verified on the display.
4. A test driver will be used to request application interrupts on a specified UTC or CDT/MET, set multiple timers, and cancel interrupt timer requests.
5. The Timer Services Stopwatch User Interface will call the Timer Services API's to start, stop, set, and clear a Stopwatch Timer. The buttons on the display will be tested and the elapsed time since the Stopwatch Timer was started will be displayed on the screen. Two or more Stopwatch User Interfaces will be brought up on the same CC&WS.
6. The Timer Services Server on the CCP will be shut down. The CDT/MET Graphical User Interface will visually indicate a loss of workstation synchronization to UTC.

1.4 Timer Services Class Diagrams